Blockchain and Federated Learning for Privacy-preserved Data Sharing in Industrial IoT

Yunlong Lu, *Student Member, IEEE*, Xiaohong Huang, *Member, IEEE*, Yueyue Dai, *Student Member, IEEE*, Sabita Maharjan, *Member, IEEE*, and Yan Zhang, *Senior Member, IEEE*

***Abstract*—The rapid increase in the volume of data generated from connected devices in Industrial Internet of Things (IIoT) paradigm, opens up new possibilities for enhancing the quality of service for the emerging applications through data sharing. However, security and privacy concerns (e.g, data leakage) are major obstacles for data providers to share their data in wireless networks. The leakage of private data can lead to serious issues beyond financial loss for the providers. In this paper, we first design a blockchain empowered secure data sharing architecture for distributed multiple parties. Then, we formulate the data sharing problem into a machine learning problem by incorporating privacy-preserved federated learning. The privacy of data is well maintained by sharing the data model instead of revealing the actual data. Finally, we integrate federated learning in the consensus process of permissioned blockchain, so that the computing work for consensus can also be used for federated training. Numerical results derived from real-world datasets show that the proposed data sharing scheme achieves good accuracy, high efficiency, and enhanced security.**

***Index Terms*—Data Sharing, Permissioned Blockchain, Federated Learning, Privacy-preserved, Industrial IoT**

I. Introduction

The amount of data generated by the connected devices in the IIoT paradigm has witnessed a massive growth in Industry 4.0. Along with the value the data brings, comes serious concerns about data privacy [1]. Data leakage may take place during data storage, data transmission and data sharing, which may lead to serious issues for both owners and providers. In this regard, existing work mainly focuses on utilizing aggregate information about the data, without breaking the privacy of the participants. They address the problem by making some modifications to the key contributions of original data, such as k-anonymity [2], l-diversity [3]. But most of the methods assume that the attackers only have limited background knowledge, where the data is still vulnerable to algorithm-based attacks or background knowledge attack. Differential privacy [4] provides the most reliable privacy guarantee, which is generally considered strong enough to protect data from privacy attacks. A Machine Learning Differentially Private (MLDP) [5] was proposed to publish data structures instead of publishing queries and answers directly, in the constraint of differential privacy.

Data from IIoT applications may include sensitive information. In this regard, protecting data privacy is a key issue. In [6], the authors proposed a protection method that satisfies differential privacy to protect location data privacy, without reducing much utility of data in Industrial IoT. There are also some works exploring the use of blockchain to enhance data security in IIoT. In [7], the authors also integrated blockchain into edge intelligence for resource allocation in IIoT. Though the combination is promising, the machine learning methods can be further improved. Therefore, some works exploited Markov models, which can illustrate activity transactions without having knowledge of the problem in hand [8], for resource allocation. For example, in [9], the authors leveraged deep reinforcement learning (DRL) for task offloading and transmission scheduling. The using of new machine learning technology also brings new security threats such as cyber stealth attacks [10], which imposes new security requirements [11] for protecting data privacy in sharing process. In [12], the authors implemented a protocol that turns a blockchain into an automated access-control manager, to ensure that users can own and control their data. In [13], the authors proposed a blockchain-enabled efficient data collection and secure sharing scheme combining Ethereum blockchain and DRL to create a reliable ad safe environment. Among these works, consensus protocols are a core technical component to achieve consensus among all participating nodes. In Proof-of-Work (PoW) [14], the miner that solves a mathematical puzzle first wins the right to produce a block. However, heavy resource requirement for solving those puzzles, limit the applicability of PoW based consensus mechanisms.

The concept of private multi-party data sharing has drawn much attention recently, as a promising approach to address the issue of computing and storage resource constraints. Several works on data sharing applications with multiple distributed data owners, have been published recently, including the sharing of horizontally partitioned data [15] and monitoring over distributed data streams [16]. There is a wide range of applications on the collaborative use of data in distributed scenarios. For instance, the authors in [17] proposed LDPGen, a multi-phase technique, to ensure local differential privacy in decentralized social graphs. Mobile Edge Computing (MEC) is a powerful technique for such resource constrained applications on distributed data. In [18], to efficiently and fairly

This work was supported by Joint Funds of National Natural Science Foundation of China and Xinjiang under Project U1603261, and National Natural Science Foundation of China under Project 61602055.

Y. Lu, X. Huang are with the Institute of Network Technology, Beijing University of Posts and Telecommunications, Beijing, China e-mail: (yunlong.lu@ieee.org; huangxh@bupt.edu.cn;).

Y. Dai is with the University of Electronic Science and Technology of China, Chengdu, China (email:daiyyue@gmail.com).

S. Maharjan is with Simula Metropolitan Center for Digital Engineering and the University of Oslo, Norway, (e-mail: sabita@simula.no).

Y. Zhang is with the University of Oslo, Norway, (e-mail:yanzhang@ieee.org).

allocating the resources in industrial IoT-based applications, the authors proposed a forward central dynamic and available approach (FCDAA) by adapting the running time of sensing and transmission processes in IoT devices. In [19], a method for conserving position confidentiality of roaming position-based services (PBSs) users was proposed based on MEC techniques in real-time industrial informatics. Recently, federated learning [20] has emerged for multiple data owners to train a global model collaboratively without sharing their raw data, respecting the privacy concerns of sharing data. In [21], the authors proposed an algorithm for client sided differential privacy-preserving federated optimization, to hide clients' contributions during the training process. Based on a hierarchical architecture in which the server aggregates the users' training updates, the authors in [22] proposed a federated learning based proactive content caching scheme.

Yet, the presence of a centralized curator in most of the existing data sharing schemes increases the risk of data leakage, especially in the application of distributed multiple parties. There are mainly two obstacles: one is, there can be a high volume of aggregated data from different parties to be processed by the curator, including some unknown fresh data; The other is, none of these parties fully trust others (including the curator), thus fearing data leakage.

To this end, the application of collaborative data sharing over distributed multiple parties in IIoT faces several challenges. New collaborative mechanisms for distributed data sharing among multiple untrusted parties, are therefore for IIoT applications. In this paper, we propose a differentially private multi-party data model sharing method based on permissioned blockchain. Instead of sharing raw data directly, we incorporate federated learning algorithms to map raw data into corresponding data models, which addresses privacy concerns in the learning phase through distributed training by local users. We also design a distributed architecture for data sharing between multiple parties based on blockchain, in which blockchain enables secure data retrieval and ensures accurate model training. Our main contributions in this paper are as follows.

- We transform the data sharing problem into a machine learning problem by leveraging federated learning to build data models and sharing the data models instead of raw data.
- We propose a new blockchain empowered collaborative architecture to share data over distributed multiple parties to reduce the risk of data leakage, through which data owners can further control the access to shared data.
- We integrate differential privacy into federated learning to further protect data privacy. We also evaluate the effectiveness of our proposed model with benchmark, open real-world datasets for data categorization.

The remainder of this paper is organized as follows. In Section II, we present our system model. In Section III, we describe permissioned blockchain and federated learning for data sharing in detail. In Section IV, we present security analysis for our proposed scheme, and provide illustrative numerical results. Finally, Section V concludes the paper and discusses the future work.

II. System Model

In this paper, we consider a common distributed data sharing scenario with multiple parties involved. Each participant owns his own data and is willing to share it. Contributors can make better use of their data by combining them together to implement a collaborative task. For example, traffic prediction can make greater progress by utilizing data from multiple sensors. An illustration of data sharing among various devices is shown in Fig. 1. Our goal is to design a secure data

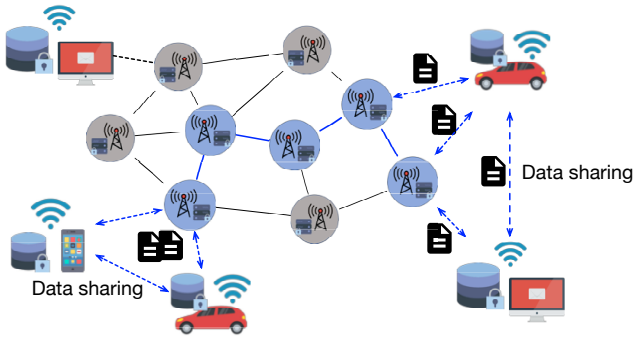


Fig. 1: An example of distributed data sharing

sharing mechanism which can share data among distributed multiple users intelligently while also maintaining data privacy effectively. We consider N parties (data holders) and a union dataset D. For any party P_i, it holds a local dataset $D_i \in D$. Each of the N parties agrees on sharing its data without revealing any private information. Let $R = r_1, r_2, ..., r_m$ be the data sharing requests with queries r_i submitted by a requester, instead of return the raw data, we provide the computed results towards these queries for sharing. Then all the participants related to the request work together with the corresponding learning algorithm to train a global model $\mathcal{M}$, without leaking any private data. Finally, the trained global data model $\mathcal{M}$ will be returned to the data recipient. Leveraging the received model, data recipients can get answers $R(\mathcal{M})$ towards their data sharing requests locally.

A. Treat Model

We focus on collaborative data sharing, where K data providers (owners) and one data requester work together to accomplish a data sharing task. The data providers and data requester are considered as dishonest. The proposed mechanism is vulnerable to three types of threats. The first is the quality of the provided data. Dishonest providers may provide biased and inaccurate results to the requester, reducing the usability of the entire shared data. The second is data privacy. Providers and receivers may try to infer the private data of others from shared data, which may lead to unwanted sensitive data leakage from data providers. The threat of collusion also exists if a group of participants try to infer the data of other participants. The third is data authority management. Once the raw data is shared, the data owner will lose control over these data and the data may be shared to other unauthorized entities by a dishonest participant.

B. Our Proposed Architecture

Our proposed data sharing architecture is shown in Fig. 2. The proposed system consists of two modules: permissioned blockchain module and federated learning module. Permissioned blockchain establishes secure connections among all the end IoT devices through its encrypted records, which is maintained by the entities equipped with computing and storage resources, named super nodes, such as Base Stations (BS) and Road Side Unites (RSUs). There are two types of transactions in our permissioned blockchain: retrieval transactions and data sharing transactions. For the privacy concerns and due to storage limitation, we use permissioned blockchain only to retrieve the related data and manage the accessibility of data, instead of to record the raw data. Moreover, permissioned blockchain records all the sharing events of data, which can trace the use of data for further audit.

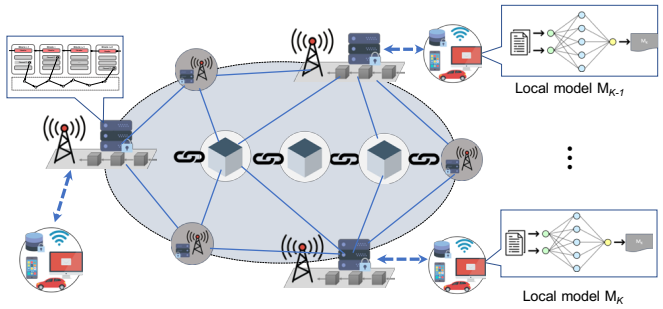


Fig. 2: Architecture of secure data sharing scheme

Suppose all the parties that agreed for data sharing have been registered in the permissioned blockchain, by uploading retrieval records to the blocks. A data requester launches a data sharing request Req containing a set of queries $F_x = \{f_1, f_2, ..., f_x\}$ to its nearby super node SN_{req}. Fig. 3 shows the working mechanism of our scheme. Nearby super node SN_{req} first searches permissioned blockchain to check whether the request has been processed before. If there is a hit, the request will be forwarded to the node that has cached the results towards request Req. The cached results are then sent to the requester as a reply. Otherwise, for a new data sharing request, the multi-party data retrieval process is executed to find the related parties according to the registration records. We regard these parties as committee nodes which are responsible for driving the consensus in permissioned blockchain. Then the committee nodes train a global data model $\mathcal{M}$ jointly by federated learning. Once the model is trained, the data requester r uses $Req = \{f_1, f_2, ..., f_x\}$ as the input of the $\mathcal{M}$ and gets the corresponding sharing results $\mathcal{M}(Req)$. Data model $\mathcal{M}$ can accept any query f_x in the query set F_x and provide a result $\mathcal{M}(f_x)$ for the query. In addition, as a machine learning model, $\mathcal{M}$ can also make predictions on the fresh query that $fy \notin F_x$.

III. Blockchain and Federated Learning for Secure Data Sharing

In this paper, we consider the problem of privacy-preserving data sharing in decentralized multiple parties. Due to the constrained resources of edge users and their privacy concerns, we share the federated data model learned over decentralized multiple parties instead of the original data. The data model contains valid information towards the requests and minimized private data of participants.

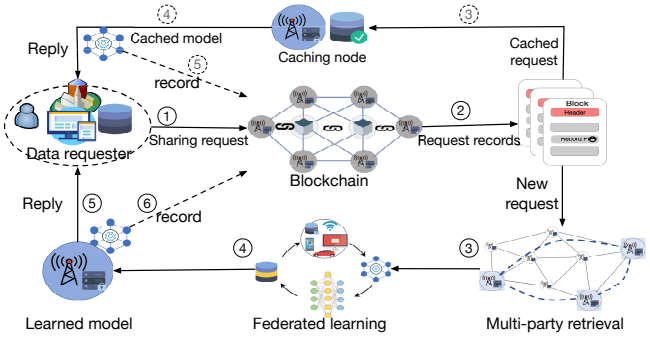


Fig. 3: The working mechanism of proposed method

A. Normalized Weighted Graph

It is challenging for the IIoT end devices to output and maintain structured data due to limited computing and storage resources. Instead, they are more likely to generate unstructured data, e.g., in the form of text files. Study regarding such data is limited. To fill this gap, we focus on the unstructured data - textual data in our data sharing scenarios. We define a two-step distance metric learning scheme for retrieval of the textual data, which can quantify the similarity of specified data.

To improve the computing and storage efficiency with constrained resources, we leverage the graphs to represent original data for further process, as defined in Definition 1, which can retain more structure and context information.

Definition 1 (Weighted Context Graphs): A weighted context graph $G = \{V, E\}$ comprises a set of nodes (key terms) V and a set of edges $E \subseteq V \times V$. Each node n_i contains a text term t_{n_i} and its weight w_i^n, (n_i, w_{n_i}). Each edge e_{ij} connects node n_i and n_j with a weight $w_{e_{ij}}$ denoting the correlation degree.

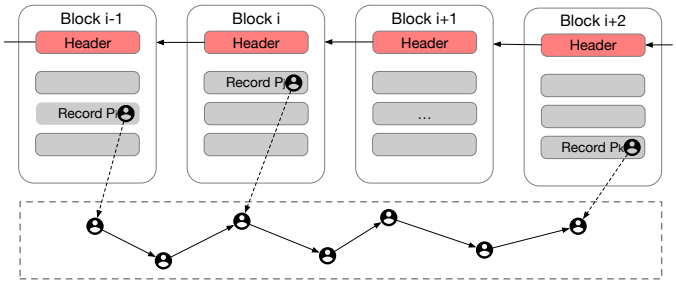
We use weight matrix $A = [a_{ij}]$ to represent the graph, where $a_{ij} = w_{n_i}$ if $i = j$ and $a_{ij} = w_{e_{ij}}$ if $i \neq j$. We leverage term frequency - inverse document frequency (TF-IDF) to construct our graphs. Thus, all the textual files are transferred into weighted context graphs $\{G_1, G_2, ..., G_n\}$.

In the second step, we serialize the graphs. Although graphs keep much context information, they are difficult to be further processed as input by machine learning algorithms. We map the graphs into liner vectors by serializing them into a sequenced vector. The graphs are first merged into a global graph $G = G_1 \cup G_2, ..., \cup G_n$. For global graph $G = \{V, E\}$, let k be the number of representative vertices. Then, the size of normalized attributes for nodes will be k and the size of normalized attributes for edges will be $k \times (k-1)/2$. Thus the normalized vector $Seq = V \cup E = \{V_1, ..., V_k\} \cup \{E_1, E_2, ..., E_{k(k-1)/2}\}$. We leverage *Jaccard similarity* [23] as the distance function to cluster documents with k-means algorithm. With the assistance of the normalized weighted graph and the defined distance

metric, we cluster dataset $\{D_1, ..., D_n\}$ into various categories according to textual similarity. We also divide the participated users into different groups according to their data.

B. Multi-party Data Retrieval

Since most of the data is sensitive and the amount of data is large, it is a resource intensive and risky task to put data on the blockchain with its limited storage space. Thus, we utilize blockchain to retrieve data, while the real data is stored locally by its owners. When a new data provider participates in, its unique identity (ID) is recorded as a transaction in the blockchain, together with the profiles of its data, including data categories, data types and data size. All the profiles of data from multiple participants will be recorded in forms of transactions, and will be verified by the blockchain nodes through adopting Merkle tree [24]. Each data sharing event is also stored in the blockchain as a transaction. The detailed forms of the two transactions - the retrieval transactions recording data profiles of permissioned participants and the data sharing transactions recording all the data sharing events - are shown in Fig. 4.



Retrieval record	Hash pointer	Provider ID	Data categories	Data type	Data Size	Timestamp
Data sharing record	Request type	Requester ID	Data categories	Data type	Data Size	Timestamp

Fig. 4: The records of blockchain

The retrieval of related participants on blockchain towards a data sharing request is a fundamental problem to be addressed in the proposed model. Since there are many participants, those who possess data related to the request, should participate in data sharing to increase the accuracy of response results. Nonetheless, the retrieval process should not break the privacy of each participant. A distributed retrieval scheme is needed to quickly locate the requested data distributed among participants, which can collaboratively response to the request.

Inspired by previous work of Kademlia [25], we design a multi-party retrieval mechanism in blockchain. All participants $\mathcal{P}$ are partitioned into various communities according to their data categories, that is, members of a community hold similar categories of data. Each community maintains a local retrieval table of $log(n)$ records towards $log(n)$ different communities. As for each node in the community, it stores the IDs of all its community members, together with the $log(log(n))$ nodes for each of its $log(n)$ closest (most related in data categories) community. In this way, the most related participants will be kept locally at the local retrieval table of P_i, as shown in Fig. 5.

We extract a list of key terms from the data of every participant as the representative features in the form of hash

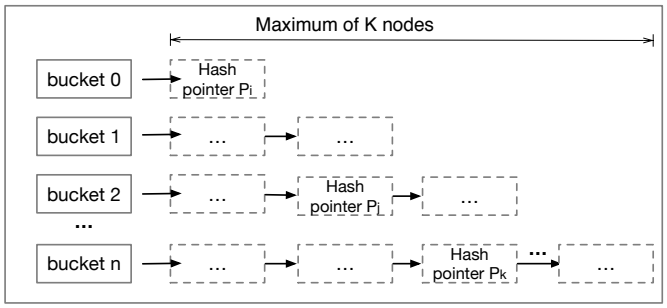


Fig. 5: The local retrieval table

values. Furthermore, due to the limited communication resource of IIoT devices, the physical distance between two nodes $d_p(P_i, P_j)$ should also be considered in the retrieval process. Then, based on Jaccard distance, the logic distance between their key terms will be

$$d_i(P_i, P_j) = \frac{\sum\limits_{m,n \in \{P_i \cup P_j - P_i \cap P_j\}} (a_{mn}^{P_i} + a_{mn}^{P_j})}{\sum\limits_{m,n \in P_i \cup P_j} (a_{mn}^{P_i} + a_{mn}^{P_j})} \cdot log(d_p(P_i, P_j)), \tag{1}$$

where $a_{mn}^{P_i}$, $a_{mn}^{P_j}$ are the elements of weighted matrix for node P_i and node P_j, respectively. The ID of each participant (device) is generated according to the logic distance. That is, the more relative two nodes are, the longer their common ID prefix will be.

Given two nodes P_i and P_j with IDs $P_i(id)$ and $P_j(id)$), respectively. The relevance distance between them is defined as

$$d(P_i, P_j) = P_i(id) \oplus P_j(id). \tag{2}$$

When a user submits a data sharing request to its nearby node P_i, all nodes in the same community with P_i send the request to the nodes in their local routing table with a certain distance to start the retrieving process. This process will be implemented recursively until all nodes within the relevant distance have been traversed. At the end of retrieval, we get the related subset nodes towards the request, $\mathcal{P}_s \subseteq \mathcal{P}$, which are also the committee nodes for running a consensus process to approve the data sharing results.

C. Data Sharing Process

Existing methods use encryption for data security. However, in data sharing scenarios, it is still risky for data holders to share the original data due to various attacks towards the encryption. A more secure method is to share the answers towards the requests, which can provide the requesters with valid information and protect privacy of data holders. The data providers share learned data models with the requesters instead of the original data.

When the data requester r initiates a sharing request Req, it submits the request to its nearby super node SN_{req} of the permissioned blockchain. SN_{req} first searches the blockchain to find out whether the request has been processed before. If there is a lookup hit, then the cached data model $\mathcal{M}$

calculated before is returned to the requester directly. Otherwise, node SN_{req} looks up the blockchain for related nodes - committee nodes , towards the sharing request r, through the aforementioned multi-party data retrieval process. The committee nodes are responsible for executing the consensus process and learn the federated data model $\mathcal{M}$ collaboratively. A committee node P_i learns a local data model m_i for the requests from requesters, then it will send model m_i to other related participants, according to the local retrieval table of P_i. This process is repeated jointly on various related parties, until all related parties are traversed. The trained data model $\mathcal{M}$ will be returned to requester r, as the answer to its data sharing requests.

The detailed steps of our data sharing scheme are as follows.

1) *Initialization:* Before a data provider P_i joins, a local clustering based on Jaccard similarity is executed to cluster its textual data into various categories. We serialize the key terms as vectors in a certain order to represent a category. The more similar two datasets are, the closer their distance is. Then its nearby super nodes, to which P_i belongs, will search the blockchain to find the records which are logically close (according to XOR distance) to it. For each participant, we generate its ID based on the hashed vectors to ensure that participants holding similar datasets have similar IDs. In addition, to enhance computing efficiency, we divide the participants in advance by running a community partition process, where all nodes are partitioned into various communities according to their distances towards each other.
2) *Registering retrieval records:* Once P_i joins, it first sends its public key PK_r and its data profiles to the nearby super node for registration. Then a data retrieval record for P_i is generated and broadcasted by the node to other nodes in permissioned blockchain for verification. Other nodes collect all received records and verify them before writing them into permissioned blockchain.
3) *Launching data sharing requests:* Data requester r posts a data sharing request $Req = \{f_1, f_2, ..., f_x\}$ to its nearby super node SN_{req}. Request Req contains the ID of r, the requested data category and the timestamp, which is signed by r with its private key SK_r.
4) *Data retrieval:* Once a nearby node receives the data sharing request, it verifies the identity of requester r. Then it searches permissioned blockchain to confirm whether the request has been processed before. If there is a record, the cached model is returned as the reply. Otherwise, it runs the multi-party retrieval process to find the related parties.
5) *Data model learning:* The related parties work collaboratively to respond to the sharing request. They run federated learning to train a global data model $\mathcal{M}$ towards the request Req. The training set is generated based on local data D and corresponding query results $fx(D)$, $D^T =< f_x, f_x(D) >$. The learning global model $\mathcal{M}$ is then returned to the requester as a reply and is cached by a node locally for future requests.
6) *Generating data sharing records:* The data sharing events between data requesters and data providers are generated as transactions and broadcasted in permissioned blockchain. All the records are collected into blocks, which are encrypted and signed by the collecting node.
7) *Carrying out consensus:* The consensus process is executed by the related nodes selected for data retrieval. Each node competes for the opportunity to write blocks to the blockchain through proof-of-work (PoW) protocol. The node who wins the competition broadcasts its block to other nodes for verification. Once the verification is passed, the block is added to the permissioned blockchain which is tamper-proof.

Combining federated learning with permissioned blockchain, the requested data can be retrieved and shared securely in industrial IoT scenarios with distributed multiple data providers, which can improve the scale and quality of shared data. However, the PoW consensus protocol incurs both high energy consumption and computation overhead, thus making it less practical for IoT devices to adopt. To address this issue, we further propose a new consensus to improve the utility and efficiency of computing work in the consensus protocol.

D. Consensus: Proof of training Quality (PoQ)

Transferring the data sharing problem into model sharing brings many benefits in data sharing. Sharing the data model only instead of original data, helps protect privacy of data owners. In addition, the machine learning data models are more effective to provide the required information for new sharing requests.

Directly using existing consensus such as PoW for data sharing either brings high cost of computing and communication resources, or makes limited additional contribution to data sharing. To address this problem, we propose a federated learning empowered consensus - Proof of training Quality (PoQ) protocol. PoQ combines data model training with the consensus process, which can make better use of the nodes' computing resources.

For a specific data sharing request, we select members of the consensus committee by retrieving the related nodes for a request in the blockchain. The committee is responsible for driving the consensus process, as well as for learning of data models for requested data. The objective of federated learning is to train a global data model $\mathcal{M}$, which can provide the valid response $\mathcal{M}(Req)$ for data sharing requests Req. Model $\mathcal{M}$ can be trained by using a series of machine learning algorithms, e.g., random tree, random forest and gradient boosting decision tree (GBDT). Once constructed, model $\mathcal{M}$ can generate the answers towards data queries, even if the queries are fresh.

1) Differentially private federated learning: To protect data privacy during multi-party decentralized learning, we incorporate a differential privacy preserved mechanism into federated learning. For two neighboring datasets D and D' with at most one different record, and a set of outcomes S, a randomized algorithm $\mathcal{A}$ achieves ϵ-differential privacy if

$$Pr[\mathcal{A}(D) \in S] \leq exp(\epsilon) \cdot Pr[\mathcal{A}(D') \in S], \tag{3}$$

where ϵ is the privacy budget. The training process over distributed data owners is shown in Fig. 6. The related parties $\{P_1, P_2, ..., P_n\}$ are selected through multi-party retrieval in blockchain. The textual data they hold is transferred into normalized graph vectors $Vec_g = \langle v_1, v_2, ..., v_k, e_{11}, e_{12}, ..., e_{kk} \rangle$. When there is a data sharing request R (including a series of queries), P_i will train a local data model m_i based on Vec_{gi} towards the request first. The Vec_{gi}, composed of sensitive terms and weights, is used to train a local model in the federated learning process. Since the local model will be shared to other participants, to protect the privacy of Vec_{gi}, we incorporate differential privacy in the learning phase to train a $\hat{m_i}$ from noised data. Then P_i will send model $\hat{m_i}$ to other participants. Once $\hat{m_i}$ is received, P_{i+1} will train a new local data model $\hat{m_{i+1}}$ based on received $\hat{m_i}$ and its local data, then broadcast $\hat{m_{i+1}}$ to other participants. The data models are trained iteratively among participants. Finally, the global data model $\mathcal{M}$ will be generated where $\mathcal{M} = \{\hat{m_1} \cup \hat{m_2} ... \cup \hat{m_n}\}$.

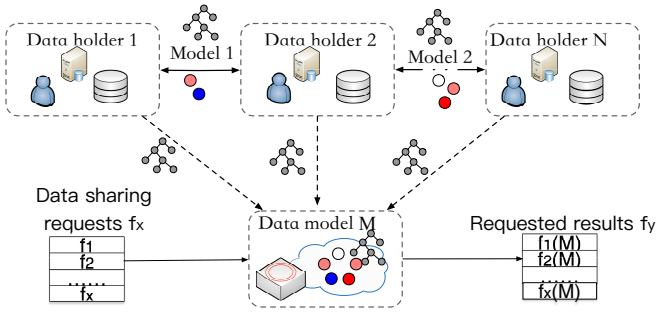


Fig. 6: The overview of training in distributed scenario

For any participant P_i, there are three steps to learn the local model $\hat{m_i}$:

- Selecting training samples: The data owner selects related data D towards the request set R and transfer them into normalized graph vectors Vec.
- Differential private local model training: The noise calibrated by sensitivity s is added to local data Vec_i . The local data model $\hat{m_i}$ is trained locally at P_i, by using machine learning algorithm on the selected noisy data Vec_i.
- Collaborative multi-party learning: The $Laplace$ mechanism is applied on local data model m_i to achieve differential privacy

$$\hat{m_i} = m_i + Laplace(s/\epsilon), \tag{4}$$

where s is the value of sensitivity, as shown in Eq. (5),

$$s = \max_{D,D'} ||f(D) - f(D')||_1. \tag{5}$$

Then the noise-added model $\hat{m_i}$ is broadcasted as a transaction of the blockchain to other participants for federated learning. This process is repeated iteratively until the performance of the federated model achieves the threshold or the training time runs out.

Algorithm 1 illustrates the overall process of our proposed scheme.

Algorithm 1 Differential private federated learning

Input: data request Req, related participants $\mathcal{P}$, iteration times $iter = 0$
Output: data model $\mathcal{M}$

for each participant $p_i \in \mathcal{P}$ **do**
  while $accuracy \leq Threshold$ **do**
    if $iter = 0$ **then**
      Construct a new differential data model $\hat{m_i}$ based on its noise-added vector data Vec_i
      Broadcast $\hat{m_i}$ to other related participants according to local retrieval tables
    else
      Construct differential private data model $\hat{m_i}$ with previously received models
      Broadcast $\hat{m_i}$ to the other participants who are engaged in the data sharing process
      $iter = iter + 1$
    end if
    $\mathcal{M} = \frac{1}{k} \sum_k \hat{m_i}$
  end while
end for
return $\mathcal{M}$ to the requester

2) Training quality based consensus: The consensus process is executed by the selected committee based on the work of collaborative training. The committee nodes are a subset of all the participants. The communication overhead is reduced by sending consensus messages only to the committee nodes instead of all the nodes. However, the reduction in the number of nodes also makes it more challenging to achieve consensus. In order to balance the overhead and the security, we provide the proof of training work for consensus in data sharing. The committee leader is selected based on the quality of the trained model. Since each committee node trains a local data model, the quality of the model should be verified and measured during the consensus process. We leverage prediction accuracy to quantify the performance of the trained local model. More specifically, in the classification during training, the accuracy is denoted by the fraction of correctly classified records. While in the task of regression, the accuracy is measured by mean absolute error (MAE):

$$MAE(m_i) = \frac{1}{n} \sum_{i=1}^{n} |y_i - f(x_i)|, \tag{6}$$

where $f(x_i)$ is the prediction value of model m_i and y_i is the real value of the records. The lower the MAE of model m_i is, the higher the accuracy of m_i will be.

After the differentially private collaborative training, we get the trained global data model $\mathcal{M}$ and local model m_i for each committee node. The consensus process is executed by the committee. During responding to a data sharing request, a committee node P_i transmits its trained model m_i to the next committee node. The transmissions are recorded as model transactions t_{m_i}, together with its $MAE(m_i)$. The record tuple is shown in Table I. The illustration of model transactions in training process is shown in Fig. 7. To encrypt and sign

TABLE I: The record tuple of a model transaction

hash pointer	owner id	receiver id	model data	MAE	timestamp

the messages, P_i has a pair of public and private keys (PK_i, SK_i). Then P_i broadcasts the encrypted transmission $E(SK_i(t_m), PK_i)$ to other committee nodes. A committee node P_j collects all model transactions and stores them locally as candidate blocks. As a proof of training work, P_j verifies all transactions it receives by calculating the MAE defined in Eq. (6). The MAE for P_j, $MAE^u(P_j)$ is calculated as

$$MAE^u(P_j) = \gamma \cdot MAE(m_j) + \frac{1}{n}\sum_i MAE(m_i), \quad (7)$$

where $MAE(m_j)$ is the MAE of the locally trained model m_j, and γ is the weight parameter denoting the contribution of P_j to the global model, decided by the training data size of P_j and other participants $\gamma = 1 + |d_j| / \sum_i |d_i|$.

When the consensus process starts, the committee node with the lowest MAE^u at that time will be elected as the committee leader through MAE-based voting. The leader is responsible for driving the consensus process among participated nodes. As mentioned earlier, the leader gathers all the transactions it received at the beginning, including the final data model $\mathcal{M}$, to form a block $B_k = (H_k, \langle t_{m_i} \rangle, \mathcal{M})$, where H_k is the header of B_k. Then the leader broadcasts B_i to all members of the committee for approval. In addition to the regular verification on a block (e.g., the header format, block size, timestamp), the committee nodes also audit the block by verifying the model transaction track, as what the verification nodes do to verify a transaction's amount in bitcoin. Each verifying node calculates the $MAE(m_i)$ for each model transaction and $MAE(\mathcal{M})$. If the calculated MAE is within a certain range, an approval will be sent to the leader. If the block containing all transactions is approved by every committee node, the leader will send the block data signed with its signature to all nodes. Then the records will be stored in the blockchain, which are tamper-proof.

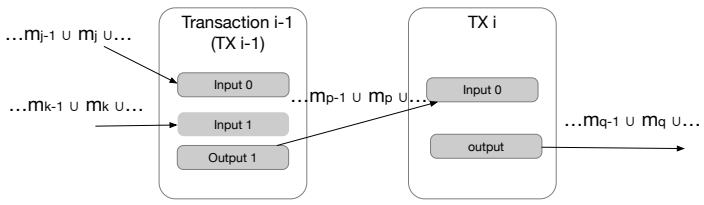


Fig. 7: Illustration of model transactions

The process for training work based consensus is illustrated in Fig. 8.

IV. Security Analysis and Numerical Results

A. Security Analysis

The use of permissioned blockchain establishes a secure mechanism for multiple parties without mutual trust. We integrate federated learning into the consensus process of permissioned blockchain to address the aforementioned security threats.

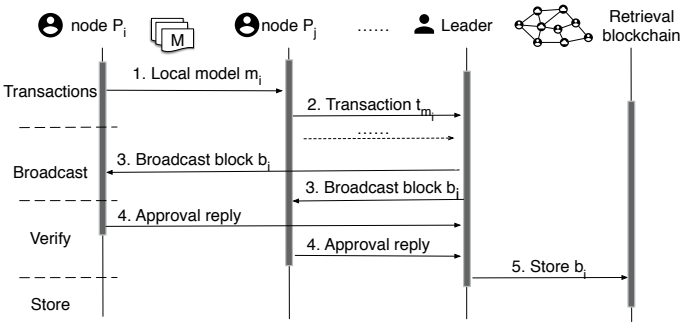


Fig. 8: The consensus process of PoQ

1) *Achieving differential privacy:* According to the definition of differential privacy, if each step of data processing follows the requirement of differential privacy, i.e., Eq. (3), the final results will satisfy differential privacy [26]. Algorithm 1 shows that the privacy budget is only consumed in step 4 where noise is added to data vectors. The other steps, using the training methods (e.g., trained tree) for calculating residuals, are only mapping operations that are data-independent and will not disclose any private information. Therefore, Algorithm 1 satisfies ϵ-differential privacy.
2) *Removing centralized trust:* The permissioned blockchain takes the place of a trusted curator to connect each participant through multi-party data retrieval. The centralized trust, which incurs high risk of data leakage, is no more required in the proposed blockchain empowered data sharing scheme.
3) *Guaranteeing the quality of shared data:* To prevent the dishonest provider from sharing invalid data, the PoQ consensus process validates the quality of learned data models by other data providers, and only the qualified models are preserved.
4) *Secure data management:* Only the data retrieval is uploaded to the permissioned blockchain while real data is stored locally by each data provider. Data owners can control the authority of their own data. Moreover, permissioned blockchain uses a series of cryptographic algorithms such as elliptic curve digital signature algorithm and asymmetric cryptography to guarantee the security of data.

B. Evaluation Setup

We conduct evaluations of the proposed secure data sharing scheme on two real-world data sets, which is widely used for evaluating text-related machine learning algorithms. The first one is Reuters dataset [27], a benchmark dataset for classification tasks. The dataset consists of a series of short files in various topics appeared on Reuters newswire. It contains a total of 15732 files in 116 categories. The second one is the 20 newsgroups data set [28], which is a collection of approximately 20000 newsgroup files. The data is partitioned into 20 different groups, where each group is related to one

topic. The data in the two datasets is unstructured short text, which is quite different from the structured data in databases. We use the two datasets to simulate the large amount of unstructured short data pieces in IIoT, such as the configuration files generated from various vehicular applications and the status log files.

We divide the sorted data set into shards and re-combine the shards into subsets to simulate the distributed multiple participants in our data sharing scheme. Classification analysis is used to simulate the data sharing tasks. We implement our improved distributed gradient boosting decision tree on textual data to execute the federated learning process over distributed data sets.

C. Numerical Results

Receiver Operating Characteristic (ROC) curve is widely used to illustrate the diagnostic ability of a classifier scheme, whose discrimination threshold is varied. We use the area under the ROC curve (AUC) to evaluate the accuracy of the model. The performance of our proposed mechanism in various distributed datasets, compared with a benchmark method, Text Graph Convolutional Networks (Text GCN) [29], is shown in Fig. 9 and Fig. 10. From Fig. 9 we can conclude that compared with the benchmark method, most of our testing groups obtain high accuracy with an average AUC value of 0.918, which indicates that our proposed federated learning mechanism achieves high diagnostic ability. We can also observe that the accuracy changes not so smoothly. The reason is that the number of files in each subset is a fixed discrete value, and the accuracy is also related to the characteristics of files from different subset, which can not be a continuously changing value as the size of data increases. Fig. 10 shows the AUC results with various number of data providers (3, 6 and 9). It can be seen that the AUC results change little as the number of data providers increases, which indicates our proposed model is scalable. Since the global model in federated learning is aggregated from all local models, and each local model is trained on local data, the number of data providers has little effect on the performance of the aggregated model. Fig. 11 shows that the running time of our proposed mechanism varies from milliseconds to seconds with an average of 880ms in different sub-datasets. From Fig. 12, we can see that, for the same dataset, the running time increases with the number of data providers involved. The reason is that the more data providers, the more time it takes to implement the collaborative working process. Moreover, the results of running time show that the performance of the proposed federated learning based data sharing scheme is near real-time.

Through the above evaluation, we can observe that the increase in data providers has little effect on the accuracy of our proposed scheme, while the running time increases evidently. The stable accuracy is due to that our scheme can execute parallel local training, where the federated mode ensures the stable learning accuracy. Yet the increase in data providers brings more local models to be updated and computed, which increases the time for training and update transmission. Thus, the running time increases with the increase of data providers. Despite the slightly increased running time, the participation of multiple data providers enlarges the scale of data for computing results, which enables the sharing of data to improve the quality of vehicular applications.

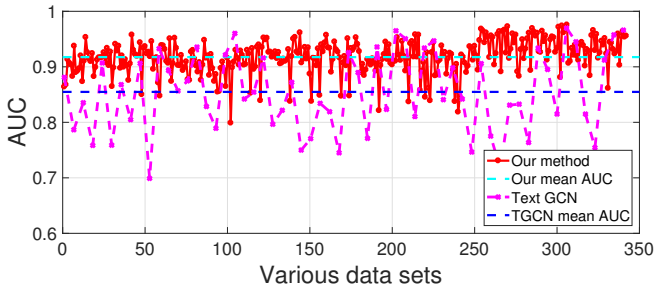


Fig. 9: AUC in various data sets (20News)

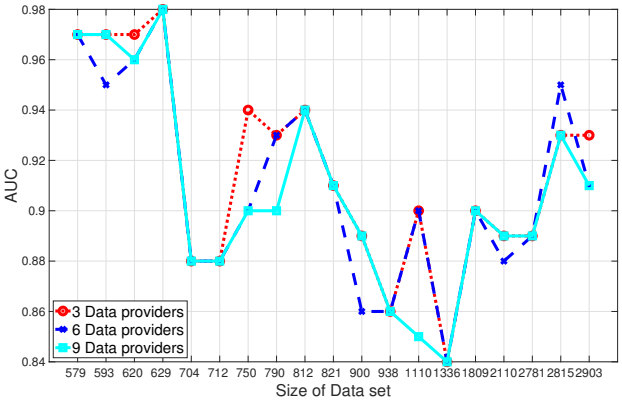


Fig. 10: AUC in various data sets (Reuters)

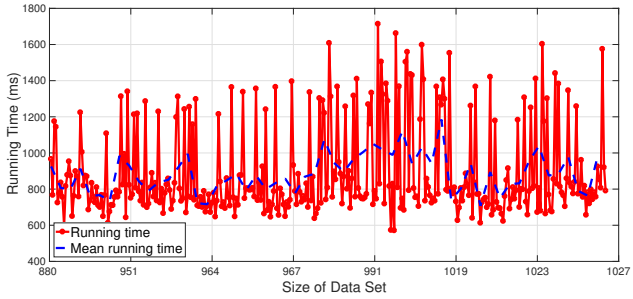


Fig. 11: Running time in various data sets (20News)

V. Conclusion

In this paper, we proposed a privacy-preserving data sharing mechanism for distributed multiple parties in industrial IoT applications, which incorporates federated learning into permissioned blockchain. The illustrative numerical results, showed that our blockchain empowered data sharing scheme enhances the security during sharing process without requiring centralized trust. Moreover, by integrating federated learning

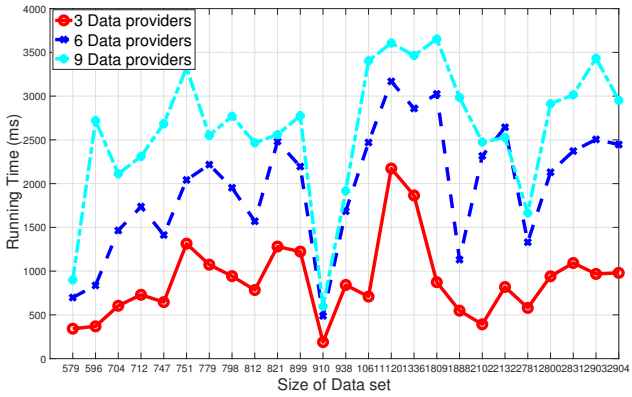


Fig. 12: Running time in various data sets (Reuters)

into the consensus process of permissioned blockchain, we not only improved the utilization of computing resource but also increased the efficiency of the data sharing scheme. Numerical results on two benchmark real-world datasets corroborated that our proposed mechanism can enable secure data sharing with high efficiency and utility.

The combination of blockchain and federated learning is a promising way to enable secure and intelligent data sharing in IIoT. However, how to efficiently guarantee data privacy by applying blockchain technique is still an open issue, which needs to be further explored by analyzing more security threats and developing more effective solutions. Moreover, how to improve the utility of data models mapped from raw data, regardless of the specific computing tasks and machine learning algorithms, is a critical problem to be addressed in data sharing. New intelligent mechanisms are required to improve data utility. In addition, the limited resource of devices imposes new challenges for improving the efficiency of data sharing in IIoT.

REFERENCES

[1] E. Sisinni, A. Saifullah, S. Han, U. Jennehag, and M. Gidlund, "Industrial internet of things: Challenges, opportunities, and directions," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 11, pp. 4724–4734, Nov 2018.

[2] L. Sweeney, "k-anonymity: A model for protecting privacy," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 10, no. 05, pp. 557–570, 2002.

[3] A. Machanavajjhala, J. Gehrke, D. Kifer, and M. Venkitasubramaniam, "l-diversity: Privacy beyond k-anonymity," in *22nd International Conference on Data Engineering (ICDE'06)*. IEEE, 2006, pp. 24–24.

[4] C. Dwork, "Differential privacy in new settings," in *Proceedings of the twenty-first annual ACM-SIAM symposium on Discrete Algorithms*. SIAM, 2010, pp. 174–183.

[5] T. Zhu, G. Li, W. Zhou, and S. Y. Philip, "Differentially private data publishing and analysis: a survey," *IEEE Transactions on Knowledge and Data Engineering*, vol. 29, no. 8, pp. 1619–1638, 2017.

[6] C. Yin, J. Xi, R. Sun, and J. Wang, "Location privacy protection based on differential privacy strategy for big data in industrial internet of things," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 8, pp. 3628–3636, Aug 2018.

[7] K. Zhang, Y. Zhu, S. Maharjan, and Y. Zhang, "Edge intelligence and blockchain empowered 5g beyond for industrial internet of things," *IEEE Network Magazine*, to be published.

[8] C. Alcaraz, L. Cazorla, and G. Fernandez, "Context-awareness using anomaly-based detectors for smart grid domains," in *International Conference on Risks and Security of Internet and Systems*. Springer, 2014, pp. 17–34.

[9] K. Zhang, S. Leng, X. Peng, P. Li, S. Maharjan, and Y. Zhang, "Artificial intelligence inspired transmission scheduling in cognitive vehicular communications and networks," *IEEE Internet of Things*, to be published.

[10] L. Cazorla, C. Alcaraz, and J. Lopez, "Cyber stealth attacks in critical information infrastructures," *IEEE Systems Journal*, vol. 12, no. 2, pp. 1778–1792, 2016.

[11] C. Alcaraz and J. Lopez, "Analysis of requirements for critical control systems," *International journal of critical infrastructure protection*, vol. 5, no. 3-4, pp. 137–145, 2012.

[12] G. Zyskind, O. Nathan, and A. Pentland, "Decentralizing privacy: Using blockchain to protect personal data," in *2015 IEEE Security and Privacy Workshops*, May 2015, pp. 180–184.

[13] C. H. Liu, Q. Lin, and S. Wen, "Blockchain-enabled data collection and sharing for industrial iot with deep reinforcement learning," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 6, pp. 3516–3526, June 2019.

[14] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," 2008.

[15] S. Goryczka, L. Xiong, and B. C. Fung, "m-privacy for collaborative data publishing," *IEEE Transactions On Knowledge And Data Engineering*, vol. 26, no. 10, pp. 2520–2533, 2014.

[16] A. Friedman, I. Sharfman, D. Keren, and A. Schuster, "Privacy-preserving distributed stream monitoring." in *NDSS*, 2014.

[17] Z. Qin, T. Yu, Y. Yang, I. Khalil, X. Xiao, and K. Ren, "Generating synthetic decentralized social graphs with local differential privacy," in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 2017, pp. 425–438.

[18] A. H. Sodhro, S. Pirbhulal, and V. H. C. de Albuquerque, "Artificial intelligence-driven mechanism for edge computing-based industrial applications," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 7, pp. 4235–4243, July 2019.

[19] A. K. Sangaiah, D. V. Medhane, T. Han, M. S. Hossain, and G. Muhammad, "Enforcing position-based confidentiality with machine learning paradigm through mobile edge computing in real-time industrial informatics," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 7, pp. 4189–4196, July 2019.

[20] J. Konečnỳ, H. B. McMahan, F. X. Yu, P. Richtárik, A. T. Suresh, and D. Bacon, "Federated learning: Strategies for improving communication efficiency," *arXiv preprint arXiv:1610.05492*, 2016.

[21] R. C. Geyer, T. Klein, and M. Nabi, "Differentially private federated learning: A client level perspective," *arXiv preprint arXiv:1712.07557*, 2017.

[22] Z. Yu, J. Hu, G. Min, H. Lu, Z. Zhao, H. Wang, and N. Georgalas, "Federated learning based proactive content caching in edge computing," in *2018 IEEE Global Communications Conference (GLOBECOM)*. IEEE, 2018, pp. 1–6.

[23] S. Niwattanakul, J. Singthongchai, E. Naenudorn, and S. Wanapu, "Using of jaccard coefficient for keywords similarity," in *Proceedings of the International MultiConference of Engineers and Computer Scientists*, vol. 1, no. 6, 2013.

[24] A. Kosba, A. Miller, E. Shi, Z. Wen, and C. Papamanthou, "Hawk: The blockchain model of cryptography and privacy-preserving smart contracts," in *2016 IEEE Symposium on Security and Privacy (SP)*, vol. 00, 2016, pp. 839–858.

[25] P. Maymounkov and D. Mazières, "Kademlia: A peer-to-peer information system based on the xor metric," in *Peer-to-Peer Systems*, P. Druschel, F. Kaashoek, and A. Rowstron, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2002, pp. 53–65.

[26] C. Dwork, "A firm foundation for private data analysis," *Communications of the ACM*, vol. 54, no. 1, pp. 86–95, 2011.

[27] D. D. Lewis, "Reuters dataset," http://www.daviddlewis.com/resources/testcollections/, 2019.

[28] "20newsgroups," http://qwone.com/~jason/20Newsgroups/, 2019.

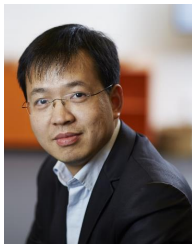
[29] L. Yao, C. Mao, and Y. Luo, "Graph convolutional networks for text classification," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, 2019, pp. 7370–7377.

Yunlong Lu received the B.S. degree in electronic information science and technology from Beijing Forestry University, Beijing, China, in 2012 and the M.S degree in School of Computer Science from Beijing University of Posts and Telecommunications (BUPT), Beijing, China, in 2015. He is currently working towards the Ph.D. degree in Computer Science and Technology with the Institute of Network Technology, BUPT, and a visiting Ph.D. student with the University of Oslo, Norway. His current research interests include blockchain, wireless networks, and privacy-preserving machine learning.

Xiaohong Huang received her B.E. degree from Beijing University of Posts and Telecommunications (BUPT), Beijing, China, in 2000 and Ph.D. degree from the school of Electrical and Electronic Engineering (EEE), Nanyang Technological University, Singapore in 2005. Since 2005, Dr. Huang has joined BUPT and now she is an associate professor and director of Network and Information Center in Institute of Network Technology of BUPT. Dr. Huang has published more than 50 academic papers in the area of WDM optical networks, IP networks and other related fields. Her current interests are Internet architecture, software defined networking, and network function virtualization.

Yueyue Dai received the B.Sc. degree in communication and information engineering from the University of Science and Technology of China (UESTC), in 2014, where she is currently pursuing the Ph.D. degree. She is now a visiting Ph.D. student with the University of Oslo, Norway. Her current research interests include wireless network, mobile edge computing, Internet of Vehicles, blockchain, and deep reinforcement learning.

Sabita Maharjan (M'09) received the Ph.D. degree in networks and distributed systems from the Simula Research Laboratory, and University of Oslo, Norway, in 2013. She is currently a Senior Research Scientist at the Simula Metropolitan Center for Digital Engineering, Norway, and an Associate Professor at the University of Oslo. Her current research interests include wireless networks, network security and resilience, smart grid communications, Internet of Things, machine-to-machine communication, software-defined wireless networking, and the Internet of Vehicles.



Yan Zhang is a Full Professor at the Department of Informatics, University of Oslo, Norway. His current research interests include: next-generation wireless networks leading to 5G Beyond, green and secure cyber-physical systems (e.g., smart grid and transport). He received a Ph.D. degree in School of Electrical & Electronics Engineering, Nanyang Technological University, Singapore. He is an Editor of several IEEE publications, including IEEE Communications Magazine, IEEE Network, IEEE Transactions on Vehicular Technology, IEEE Transactions on Industrial Informatics, IEEE Transactions on Green Communications and Networking, IEEE Communications Surveys & Tutorials, IEEE Internet of Things, IEEE Systems Journal and IEEE Vehicular Technology Magazine. He serves as chair positions in a number of conferences, including IEEE GLOBECOM 2017, IEEE PIMRC 2016, and IEEE SmartGridComm 2015. He is IEEE VTS (Vehicular Technology Society) Distinguished Lecturer. He is a Fellow of IET. He serves as the Chair of IEEE ComSoc TCGCC (Technical Committee on Green Communications & Computing). He received the award 2018 "Highly Cited Researcher" according to Clarivate Analytics.